第六节 监控管理

JMX 规范

基本概念

ObjectName

例如：Tomcat:type=ThreadPool,name="http-bio-8080"

JMX MBean 属性

类同于 Java 对象属性（JavaBeans Property，Readable，Writeable）

JMX API - `MBeanAttributeInfo`

JavaBeans API - `PropertyDescriptor`

JMX MBean 操作

类同于 JavaBeans Method

JMX API - `MBeanOperationInfo`

JavaBeans API - `MethodDescriptor`

JMX MBean 通知

类似于 JavaBeans 事件

标准 MBean

在命名规则方面，标准 MBean 接口必须是 Java interface 之外，其接口名称必须以 "MBean" 为后缀，如

com.example.MyClassMBean ，同时，其实现类的名称必须是 "MyClass"（所在的 package 没有限制），与

Java Beans 内省类似。

JMX 内省 -> 将预定 MBean 接口进行解析元信息 -> MBeanInfo

- 属性 - MBeanAttributeInfo
- 操作 - MBeanOperationInfo
- 构造器 - MBeanConstructorInfo

- 通知 - MBeanNotificationInfo

JavaBeans 内省 -> 将预定类型进行解析元信息 -> BeanInfo

- 属性 - PropertyDescriptor
- 方法 - MethodDescriptor
- 构造器 -
- 监听器 - EventSetDescriptor

动态 MBean

javax.management.DynamicMBean

开放 MBean

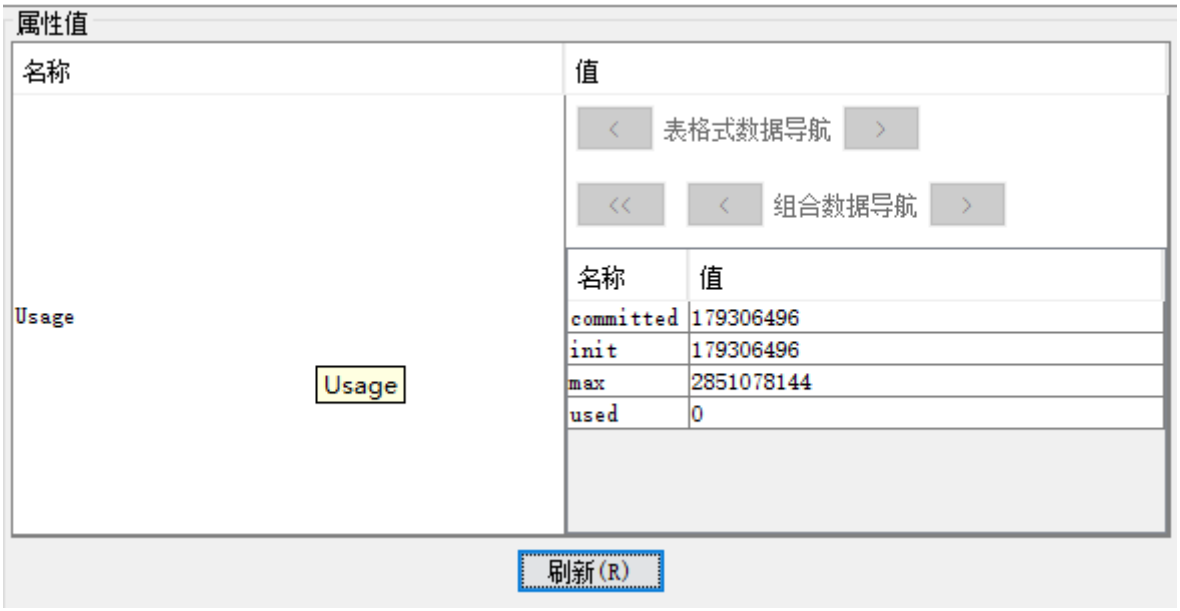
```
{
    id:1,
    name:"小马哥",
    ...
    phone_number: "123456789", // ->
phoneNumber
}
```

```
public class MemoryUsage {
    private final long init; // -Xms
    private final long used;
    private final long committed;
    private final long max; //  -Xmx
    ...
}
```

javax.management.openmbean.OpenType

- 简单类型
  - 原型类型（9种）+ 包装类型 + Void
  - java.math.BigDecimal
  - java.math.BigInteger
  - java.util.Date
  - javax.management.ObjectName
- 合成类型 - CompositeData - 单一类型（类似于类结构）
- 扁平类型 - TabularData - 集合类型（数组、List、Set、Queue、Map 等）

对应的类型是 - javax.management.openmbean.CompositeData



开放类型中，CompositeData -> 键值对

类的结构是属性

扁平结构只有单一维度

List -> elementType -> Integer

```
public class User {  // User -> CompositeData
    private Long id;

    private List<Address> addresses; //
addresses -> TabularData
    // Address -> CompositeData
}
```

```
{
    id: 1L,
    addresses:[{},{}]
}
```

模型 MBean

总结

Java 需要一套统一的近程和远程监控和管理的架构 -> JMX

三个层次

- 装配层 - 将被管理资源（对象）封装成 JMX 认可的结构，标准 MBean、动态 MBean、开放 MBean 和模型 MBean
- 代理层 - JMX 注册装配层资源的服务器，并且能够暴露成相关通讯协议给分布式服务层
- 分布式服务层 - 通过 JMX 连接和协议适配来实现的应用客户端

四种 MBean 类型

- 标准 MBean（静态编译时结构）
- 动态 MBean = DynamicMBean（动态运行时结构）
- 开放 MBean = DynamicMBean（动态运行时结构）

- 模型 MBean = DynamicMBean（动态运行时结构）

从底层而言，所有类型 MBean 最终会变成 DynamicMBean

com.sun.jmx.mbeanserver.Introspector#makeDynamicMBean 方法决定将 DynamicMBean 直接返回，将 标准 MBean 转换成 DynamicMBean。

五种结构

描述性接口（元信息接口），类似于 JavaBeans 自省

- 属性
- 操作
- 通知
- 构造器
- 参数

- JDK 1.5 引入 JMX
- JMX 装配层存在自定义模型 -> JMX 客户端可能不存在这样的模型，因此，引入 OpenType 开放 MBean
- JMX 装配层觉得实现 MBean 或 MXBean 接口很繁杂，需要通过 POJO 注册 MBean，因此，引入模型 MBean
- JDK 1.6 引入 JConsole 与 JMX 共享 JRE -> 共享 JMX API
- Spring JMX 集中处理了 MBean、MXBean -> DynamicMBean 和 ModelMBean

相关技术

IDL 的类型规约

Protobuffer

WebServices - SOAP

XML

JSON

相关资料

Tomcat 关于 JMX 监控和管理 - http://tomcat.apache.org/tomcat-7.0-doc/monitoring.html

本周作业

- 需求一（必须）
  - 整合 https://jolokia.org/
    - 实现一个自定义 JMX MBean，通过 Jolokia 做 Servlet 代理
- 需求二（选做）
  - 继续完成 Microprofile config API 中的实现

        - 扩展 org.eclipse.microprofile.config.spi.ConfigSource 实现，包括 OS 环境变量，以及本地配置文件
        - 扩展 org.eclipse.microprofile.config.spi.Converter 实现，提供 String 类型到简单类型
    - 通过 org.eclipse.microprofile.config.Config 读取当前应用名称
        - 应用名称 property name = "application.name"